UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/781,867	02/20/2004	Edward Colles Nevill	550-513	5181

23117 7590 11/30/2011

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
PATEL, KAUSHIKKUMAR M	
ART UNIT	**PAPER NUMBER**
2186	

MAIL DATE	DELIVERY MODE
11/30/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte EDWARD COLLES NEVILL

Appeal 2009-011001
Application 10/781,867
Technology 2100

Before MAHSHID D. SAADAT, ROBERT E. NAPPI,
and BRUCE R. WINSOR, *Administrative Patent Judges*.

SAADAT, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellant appeals under 35 U.S.C. § 134(a) from the final rejection of claims 1-33, which constitute all the claims pending in this application. We have jurisdiction under 35 U.S.C. § 6(b).

We reverse.

STATEMENT OF THE CASE

Introduction

Appellant's invention relates to memory recycling process or "garbage collection" in computer systems (Spec. 2:6-7, 6:13 – 7:10).

Exemplary Claims

Claims 1 and 31 are illustrative of the invention and read as follows.

> 1. A method of controlling execution of a processing task within a data processing system, said method comprising the steps of:
>
> executing said processing task including allocating memory areas for data storage; and
>
> suspending an actual execution path of said processing task at an execution point to perform memory management, said memory management comprising the steps of:
>
> identifying at least one data item roots occurring in the course of execution and accessible to said processing task at said execution point which specify reference values pointing to respective ones of said memory areas;
>
> determining a correlation between reference values corresponding to said at least one data item roots and memory areas allocated during said execution up to said execution point by identifying at least one data item reachable from said at least one data item roots; and
>
> performing a memory management operation on allocated memory areas in dependence upon said correlation.

31. A method of identifying for a memory management operation at least one data item root and at least one data item reachable from said data item root comprising the steps of:

scanning a plurality of program instructions corresponding to said processing task and logging a data type for each store instruction corresponding to each of said at least one data item;

categorizing at least one of said at least one data item root or said at least one data item as a multiple-type variable if different data types are logged for different store instructions for a respective data item;

simulating all possible execution paths up to said execution point for each of said at least one data item root or said at least one data item categorized as a multiple-type variable and determining the data type associated with each multiple-type variable at each of said plurality of program instructions for each of said possible execution paths; and

checking said determined data type for each of said multiple-type variables at one of said plurality of program instructions corresponding to said current execution point,

wherein said memory management operation is performed in dependence upon a result of said step of checking said determined data type.

Claim Rejections

The Examiner relies on the following prior art in rejecting the claims:

Wilson, Paul R., "*Uniprocessor garbage collection techniques*," Proc. of International Workshop on Memory Management in Lecture Notes on Computer Science, Springer-Verlag., Volume 637, September 1992, pp. 1-67, accessed on http://citeseer.ist.psu.edu/wilson92uniprocessor.html.

Hosoya et al., "*Garbage Collection via Dynamic Type Inference - A Formal Treatment-*," Types in Compilation: Second International Workshop in Lecture Notes in Computer Science, Springer Berlin/Heidelberg, Volume 1473, March 1998, pp. 215-239.

Claims 1-6, 9, 11-16, 19, 21-26, and 29 stand rejected under 35 U.S.C. § 102(b) as anticipated by Wilson.

Claims 10, 20, and 30 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Wilson.

Claims 7, 8, 17, 18, 27, 28, and 31-33 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Wilson in view of Hosoya.

ISSUES

Did the Examiner err in rejecting the claims under 35 U.S.C. §§ 102(b) and 103(a)? The issues specifically turn on whether Wilson (1) discloses a memory management system that includes "suspending an actual execution path of said processing task at an execution point to perform memory management," as recited in claim 1; (2) suggests identifying at least one data item root at "said execution point," as recited in claim 31.

ANALYSIS

The Examiner relies on section 2.2 on page 9 of Wilson for disclosing the recited features of claim 1 (Ans. 3-4). Appellant contends that Wilson relates to a well-known "mark-sweep" technique (App. Br. 13-14) where live objects from the garbage are marked and traced by starting a root set (App. Br. 15). Appellant further argues that such steps are performed prior to execution of the processing task instead of suspending the processing task at an execution point, as required by claims 1 and 31 (*id.*).

The Examiner further takes the position that Wilson discloses the claimed subject matter because the reference explains the "mark-sweep" technique as an algorithm and is performed during the execution of the

program (Ans. 9). The Examiner also asserts that suspending an actual execution path at some point during a garbage collection operation is inherent and identifies Gupta[1] as the evidence in support of inherency (Ans. 9-10). The Examiner asserts that Gupta describes a mark-sweep garbage collection process known as "mostly concurrent" which has a pause or suspension (Ans. 10).

We agree with Appellant's position (Reply Br. 3) that reliance on Gupta does not indicate that Wilson's process includes any steps related to "suspending an actual execution path of said processing task at an execution point" in order to perform the memory management task. In other words, Gupta does not show that Wilson's garbage collection process necessarily includes a pause and inherently teaches the claimed step of suspending an actual execution path. Additionally, as pointed out by Appellant (Reply Br. 5-6), even if Gupta includes a pause, the Examiner has not pointed to any disclosure in Gupta related to suspending the execution path in order to accomplish the task of memory management including the recited identifying, determining, and performing steps. As such, the applied prior art does not teach or suggest the claimed step of suspending an actual execution path comprising identifying a data root the specific value pointing to a memory area, determining a correlation between reference values, or performing the memory management.

---

[1] Gupta et al., "*Turbo-charging Java HotSpot Virtual Machine, v1.4.x to Improve the Performance and Scalability of Application Servers*", Sun Microsystems, Inc., available at archive.org at http:lljava.sun.com/developerltechnicalArticles/Programming/turbo/, accessed Dec. 08, 2003, pp.1-17.

CONCLUSIONS

On the record before us, we conclude that the Examiner erred in rejecting claim 1, and other independent claims 11 and 21 which include similar limitations because Wilson does not disclose suspending an actual execution path comprising identifying a data root the specific value pointing to a memory area, determining a correlation between reference values, or performing the memory management. Similarly, the mark and sweep method of Wilson which manages memory prior to execution of the processing task, even if combined with Hosoya, is not shown to be performed at the execution point where the data root is identified, as recited in claims 31-33. Further, the Examiner has not identified any teachings in Hosoya to cure the above-mentioned deficiency of Wilson.

Therefore, we do not sustain any of the 35 U.S.C. § 102 or § 103 rejections of claims 1, 11, 21, and 31-33, or of claims 2-10, 12-20, and 22-30 dependent therefrom, over Wilson alone or in combination with Hosoya.

DECISION

The decision of the Examiner rejecting claims 1-33 is reversed.

REVERSED

tj